Computational Visual Media
DOI 10.1007/s41095-016-0068-y Vol. 3, No. 1, March 2017, 33–47

Research Article

Robust facial landmark detection and tracking across poses and expressions for in-the-wild monocular video

Shuang Liu[1], **Yongqiang Zhang**[2] (✉), **Xiaosong Yang**[1], **Daming Shi**[2], **and Jian J. Zhang**[1]

Abstract We present a novel approach for automatically detecting and tracking facial landmarks across poses and expressions from in-the-wild monocular video data, e.g., YouTube videos and smartphone recordings. Our method does not require any calibration or manual adjustment for new individual input videos or actors. Firstly, we propose a method of robust 2D facial landmark detection across poses, by combining shape-face canonical-correlation analysis with a global supervised descent method. Since 2D regression-based methods are sensitive to unstable initialization, and the temporal and spatial coherence of videos is ignored, we utilize a coarse-to-dense 3D facial expression reconstruction method to refine the 2D landmarks. On one side, we employ an in-the-wild method to extract the coarse reconstruction result and its corresponding texture using the detected sparse facial landmarks, followed by robust pose, expression, and identity estimation. On the other side, to obtain dense reconstruction results, we give a face tracking flow method that corrects coarse reconstruction results and tracks weakly textured areas; this is used to iteratively update the coarse face model. Finally, a dense reconstruction result is estimated after it converges. Extensive experiments on a variety of video sequences recorded by ourselves or downloaded from YouTube show the results of facial landmark detection and tracking under various lighting conditions, for various head poses and facial expressions. The overall performance and a comparison with state-of-art methods demonstrate the robustness and effectiveness of our method.

Keywords face tracking; facial reconstruction; landmark detection

1 Introduction

Facial landmark detection and tracking is widely used for creating realistic face animations of virtual actors for applications in computer animation, film, and video games. Creation of convincing facial animation is a challenging task due to the highly nonrigid nature of the face and the complexity of detecting and tracking the facial landmarks accurately and efficiently in uncontrolled environments. It involves facial deformation and fine-grained details. In addition, the *uncanny valley* effect [1] indicates that people are extremely capable of identifying subtle artifacts in facial appearance. Hence, animators need to make a tremendous amount of effort to localize high quality facial landmarks. To reduce the amount of manual labor, an ideal face capture solution should automatically provide the facial shape (landmarks) with high performance given reasonable quality input videos.

As a key role in facial performance capture, robust facial landmark detection across poses is still a hard problem. Typical generative models including active shape models [2], active appearance models [3], and their extensions [4–6] mitigate the influence of illumination and pose, but tend to fail when used *in the wild*. Recently, discriminative models have shown promising performance for robust facial landmark detection, represented by cascaded regression-based

[1] Bournemouth University, Poole, BH12 5BB, UK. E-mail: S. Liu, sliu@bournemouth.ac.uk; X. Yang, xyang@bournemouth.ac.uk; J. J. Zhang, jzhang@bournemouth.ac.uk.

[2] Harbin Institute of Technology, Harbin, 150001, China. E-mail: Y. Zhang, seekever@foxmail.com (✉); D. Shi, damingshi@hotmail.com.

Manuscript received: 2016-09-04; accepted: 2016-12-20

methods, e.g., explicit shape regression [7], and the supervised descent method [8]. Many recent works following the cascaded regression framework consider how to improve efficiency [9, 10] and accuracy, taking into account variations in pose, expression, lighting, and partial occlusion [11, 12]. Although previous works have produced remarkable results on nearly frontal facial landmark detection, it is still not easy to locate landmarks across a large range of poses under uncontrolled conditions. A few recent works [13–15] have started to consider multi-pose landmark detection, and can deal with small variations in pose. How to solve the multiple local minima issue caused by large differences in pose is our concern.

On the other hand, facial landmark detection and tracking can benefit from reconstructed 3D face geometry based on existing 3D facial expression databases. Remarkably, Cao et al. [16] extended the 3D dynamic expression model to work with even monocular video, with improved performance of facial landmark detection and tracking. Their methods work well with indoor videos for a range of expressions, but tend to fail for videos captured *in the wild* (ITW) due to uncontrollable lighting, varying backgrounds, and partial occlusions. Many researchers have made great efforts on dealing with ITW situations and have achieved many successes [16–18]. However, the expressiveness of captured facial landmarks from these ITW approaches is limited since most pay little attention to very useful details not represented by sparse landmarks. Additionally, optical flow methods have been applied to track facial landmarks [19]. Such a method can take advantage of fine-grained detail, down to pixel level. However, it is sensitive to shadows, light variations, and occlusion, which makes it difficult to apply in noisy uncontrolled environments.

To this end, we have designed a new ITW facial landmark detection and tracking method that employs optical flow to enhance the expressiveness of captured facial landmarks. A flowchart of our work is shown in Fig. 1. First, we use a robust 2D facial landmark detection method which combines canonical correlation analysis (CCA) with a global supervised descent method (SDM). Then we improve the stability and accuracy of the landmarks by reconstructing 3D face geometry in a coarse to dense manner. We employ an ITW method to extract a coarse reconstruction and corresponding texture via sparse landmark detection, identity, and expression estimation. Then, we use a face tracking flow method that exploits the coarsely reconstructed model to correct inaccurate tracking and recover details of the weakly textured area, which is used to iteratively update the face model. Finally, after convergence, a dense reconstruction is estimated, thus boosting the tracked landmark result. Our contributions are three fold:

- A novel robust 2D facial landmark detection method which works across a range of poses, based on combining shape-face CCA with SDM.
- A novel 3D facial optical flow tracking method for

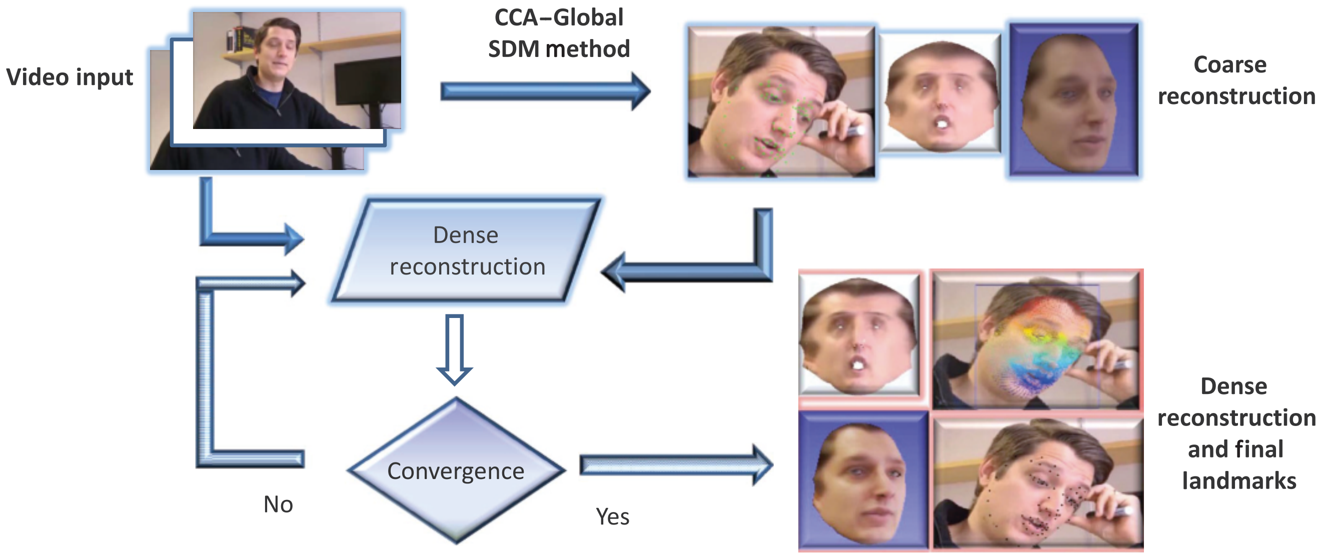


Fig. 1 Flowchart of our method.

robustly tracking expressive facial landmarks to enhance the location result.

- Accurate and smooth landmark tracking result sequences due to simultaneously registering the 3D facial shape model in a coarse-to-dense manner.

The rest of the paper is structured as follows. The following section reviews related work. In Section 3, we introduce how we detect 2D landmarks from monocular video and create the coarsely reconstructed landmarks. Section 4 describes how we refine landmarks by use of optical flow to achieve a dense reconstruction result.

2 Literature review

To reconstruct the 3D geometry of the face, facial landmarks first have to be detected. Most facial landmark detection methods can be categorized into three groups: constrained local methods [20, 21], active appearance models (AAM) [3, 22, 23], and regressors [24–26]. The performance of constrained local methods is limited in the wild because of the limited discriminative power of their local experts. Since the input is uncontrolled in ITW videos, person specific facial landmark detection methods such as AAM are inappropriate. AAM methods explicitly minimize the difference between the synthesized face image and the real image, and are able to produce stable landmark detection results for videos in controlled environments. However, conventional wisdom states that their inherent facial texture appearance models are not powerful enough for ITW problems. Although in recent literature [18] efforts have been made to address this problem, superior results to other ITW methods have not been achieved. Regressor-based methods, on the other hand, work well in the face of ITW problems and are robust [27], efficient [28], and accurate [24, 29].

Most ITW landmark detection methods were originally designed for processing single images instead of videos [8, 24, 30]. On image facial landmark detection datasets such as 300-W [31], Helen [32], and LFW [33], existing ITW methods have achieved varying levels of success. Although they provide accurate landmarks for individual images, they do not produce temporally or spatially coherent results because they are sensitive to the bounding box provided by face detector. ITW methods can only produce semantically correct but inconsistent landmarks, and while these facial landmarks might seem accurate when examined individually, they are poor in weakly textured areas such as around the face contour or where a higher level of detail is required to generate convincing animation. One could use sequence smoothing techniques as post processing [16, 17], but this can lead to an oversmoothed sequence with a loss of facial performance expressiveness and detail.

It is only recently that an ITW video dataset [34] was introduced to benchmark landmark detection in continuous ITW videos. Nevertheless, the number of facial landmarks defined in Ref. [34] is limited and does not allow us to reconstruct the person's nose and eyebrow shape. Since we aim to robustly locate facial landmarks from ITW videos, we collected a new dataset by downloading YouTube videos and recording video with smartphones, as a basis for comparing our method to other existing methods.

In terms of 3D facial geometry reconstruction for the refinement of landmarks, recently there has been an increasing amount of research based on 2D images and videos [19, 35–41]. In order to accurately track facial landmarks, it is important to first reconstruct face geometry. Due to the lack of depth information in images and videos, most methods rely on *blendshape* priors to model nonrigid deformation while structure-from-motion, photometric stereo, or other methods [42] are used to account for unseen variation [36, 38] or details [19, 37].

Due to the nonrigidness of the face and depth ambiguity in 2D images, 3D facial priors are often needed for initializing 3D poses and to provide regularization. Nowadays consumer grade depth sensors such as Kinect have been proven successful, and many methods [43–45] have been introduced to refine its noisy output and generate high quality facial scans of the kind which used to require high end devices such as laser scanners [46]. In this paper we use the FaceWarehouse [43] as our 3D facial prior. Existing methods can be grouped into two categories. One group aims to robustly deliver coarse results, while the other one aims to recover fine-grained details. For example, methods such as those in Refs. [19, 37, 40] can reconstruct details such as wrinkles, and track subtle facial movements, but are affected by shadows and occlusions. Robust methods such as Refs. [35, 36, 39] can track facial performance in the presence of noise but often miss subtle details such as small eyelid and mouth movements, which are important in conveying the target's emotion and to generate convincing animation. Although we use a 3D optical flow approach similar to that in Ref. [19] to track facial performance, we also deliver stable results even in noisy situations or when the quality of the automatically reconstructed coarse model is poor.

3 Coarse landmark detection and reconstruction

An example of coarse landmark detection and reconstruction is shown in Fig. 2. To initialize our method, we build an average shape model from the input video. First, we run a face detector [47] on the input video to be tracked. Due to the uncontrolled nature of the input video, it might fail in challenging frames. In addition to filtering out failed frames, we also detect the blurriness of remaining ones by thresholding the standard deviation of their Laplacian filtered results. Failed and blurry frames are not used in coarse reconstruction as they can contaminate the reconstructed average shape.

3.1 Robust 2D facial landmark detection

Next, inspired by Refs. [28, 48], we use our robust 2D facial landmark detector which combines shape-face CCA and global SDM. It is trained on a large multi-pose, multi-expression face dataset, FaceWarehouse [16], to locate the position of 74 fiducial points. Note that our detector is robust in the wild because the input videos for shape model reconstruction are from uncontrolled environments.

Using SDM, for one image $\boldsymbol{d}$, the locations of p landmarks $\vec{x} = [x_1, y_1, \ldots, x_p, y_p]$ are given by a feature mapping function $\vec{h}(\boldsymbol{d}(\vec{x}))$, where $\boldsymbol{d}(\vec{x})$ indexes landmarks in the image $\boldsymbol{d}$. The facial landmark detection problem can be regarded as an optimization problem:

$$f(\vec{x}_0 + \Delta\vec{x}) = \|\vec{h}(\boldsymbol{d}(\vec{x}_0 + \Delta\vec{x})) - \phi_*\|_2^2 \tag{1}$$

where $\phi_* = \vec{h}(\boldsymbol{d}(\vec{x}_*))$ represents the feature extracted according to correct landmarks $\vec{x}_*$, which is known in the training images, but unknown in the test images. A general descent mapping can be learned from training dataset. The supervised descent method form is

$$\vec{x}_k = \vec{x}_{k-1} - \boldsymbol{R}_{k-1}(\phi_{k-1} - \phi_*) \tag{2}$$

Since ϕ_* for a test image is unknown but constant, SDM modifies the objective to align with respect to the average of $\overline{\phi}_*$ over the training set, and the update rule is then modified:

$$\Delta\vec{x} = \boldsymbol{R}_k(\overline{\phi}_* - \phi_k) \tag{3}$$

Instead of learning only one $\boldsymbol{R}_k$ over all samples during one updating step, the global SDM learns a series of $\boldsymbol{R}^t$, each for a subset of samples S^t, where the whole set of samples is divided into T subsets $S = \{S^t\}_1^T$.

A generic descent method exists under these two conditions: (i) $\boldsymbol{R}\vec{h}(\vec{x})$ is a strictly locally monotone operator anchored at the optimal solution, and (ii) $\vec{h}(\vec{x})$ is locally Lipschitz continuous anchored at $\vec{x}_*$. For a function with only one minimum, these normally hold. But a complicated function may have several local minima in a relatively small neighborhood, so the original SDM tends to average conflicting gradient directions. Instead, the global SDM ensures that if the samples are properly partitioned into subsets, there is a descent method in each of the subsets. $\boldsymbol{R}_t$ for subset S_t can be solved as a constrained optimization problem:

$$\min_{S,\boldsymbol{R}} \sum_{t=1}^{T} \sum_{i\in S^t} \|\Delta\vec{x}_* - \boldsymbol{R}_t \Delta\phi^{i,t}\|^2 \tag{4}$$

$$\text{such that} \ \ \Delta\vec{x}_*^i \boldsymbol{R}_t \Delta\phi^{i,t} > 0, \quad \forall\, t, i \in S^t \tag{5}$$

where $\Delta\vec{x}_*^i = \vec{x}_*^i - \vec{x}_k^i$, $\Delta\phi^{i,t} = \overline{\phi}_*^t - \phi^i$, and where $\overline{\phi}_*^t$ averages all ϕ_* over the subset S^t. Equation (5) guarantees that the solution satisfies descent method condition (i). It is NP-hard to solve Eq. (4), so we use a deterministic scheme to approximate the solution. A set of sufficient conditions for Eq. (5) is given:

$$\Delta\vec{x}_*^{i\mathrm{T}} \Delta\boldsymbol{X}_*^t > \vec{0}, \quad \forall\, t, i \in S^t \tag{6}$$

$$\Delta\boldsymbol{\Phi}^{t\,\mathrm{T}} \Delta\phi^{i,t} > \vec{0}, \quad \forall\, t, i \in S^t \tag{7}$$

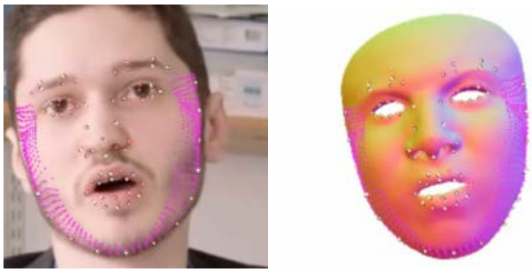


Fig. 2 Example of detected coarse landmarks and reconstructed facial mesh for a single frame.

where $\Delta \boldsymbol{X}_*^t = [\Delta\vec{x}_*^{1,t}, \dots, \Delta\vec{x}_*^{i,t}, \dots]$, each column is $\Delta\vec{x}_*^{i,t}$ from the subset S^t; $\Delta\boldsymbol{\Phi}^t = [\Delta\phi^{1,t}, \dots, \Delta\phi^{i,t}, \dots]$, and each column is $\Delta\phi^{i,t}$ from the subset S^t.

It is known that $\Delta\vec{x}$ and $\Delta\phi$ are embedded in a lower dimensional manifold for human faces, so dimension reduction methods (e.g., PCA) on the whole training set $\Delta\vec{x}$ and $\Delta\phi$ can be used for approximation. The global SDM authors project $\Delta\vec{x}$ onto the subspace spanned by the first two components of the $\Delta\vec{x}$ space, and project $\Delta\phi$ onto the subspace spanned by the first component of the $\Delta\phi$ space. Thus, there are 2^{2+1} subsets in their work. This is a very naive scheme and unsuitable for face alignment. Correlation-based dimension reduction theory can be introduced to develop a more practical and efficient strategy for low dimensional approximation of the high dimensional partition problem.

Considering the low dimensional manifold, the $\Delta\vec{x}$ space and $\Delta\phi$ space can be projected onto a medium-low dimensional space with projection matrices $\boldsymbol{Q}$ and $\boldsymbol{P}$, respectively, which keeps the projected vectors $\vec{v} = \boldsymbol{Q}\Delta\vec{x}$, $\vec{u} = \boldsymbol{P}\Delta\phi$ sufficiently correlated: (i) $\vec{v}$, $\vec{u}$ lie in the same low dimensional space, and (ii) for each jth dimension, $\operatorname{sign}(v_j, u_j) = 1$. If the projection satisfies these two conditions, the projected samples $\{\vec{u}^i, \vec{v}^i\}$ can be partitioned into different hyperoctants in this space simply according to the signs of $\vec{u}^i$, due to condition (ii). Since samples in a hyperoctant are sufficiently close to each other, this partition can carry small neighborhoods better. It is also a compact low dimensional approximation of the high dimensional hyperoctant-based partition strategy in both $\Delta\vec{x}$ space and $\Delta\phi$ space, which is a sufficient condition for the existence of a generic descent method, as mentioned above.

For convenience, we re-denote $\Delta\vec{x}$ as $\vec{y} \in \Re^n$, re-denote $\Delta\phi$ as $\vec{x} \in \Re^m$, $\boldsymbol{Y}_{s\times n} = [\vec{y}^1, \dots, \vec{y}^i, \dots, \vec{y}^s]$ collects all $\vec{y}^i$ from the training set, and $\boldsymbol{X}_{s\times m} = [\vec{x}^1, \dots, \vec{x}^i, \dots, \vec{x}^s]$ collects all $\vec{x}^i$ from the training set. The projection matrices are:

$$\boldsymbol{Q}_{r\times n} = [\vec{q}_1, \dots, \vec{q}_j, \dots, \vec{q}_r]^{\mathrm{T}}, \quad \vec{q}_j \in \Re^n$$
$$\boldsymbol{P}_{r\times m} = [\vec{p}_1, \dots, \vec{p}_j, \dots, \vec{p}_r]^{\mathrm{T}}, \quad \vec{p}_j \in \Re^m$$

The projection vectors are $\vec{v} = \boldsymbol{Q}\vec{y}$ and $\vec{u} = \boldsymbol{P}\vec{x}$. We denote the projection vectors along the sample space by $\vec{w}_j = \boldsymbol{Y}\vec{q}_j = [v_j^1, \dots, v_j^i, \dots, v_j^s]^{\mathrm{T}}$, and $\vec{z}_j = \boldsymbol{X}\vec{p}_j = [u_j^1, \dots, u_j^i, \dots, u_j^s]^{\mathrm{T}}$. This problem can be formulated as a constrained optimization problem:

$$\min_{\boldsymbol{P},\boldsymbol{Q}} \sum_{j=1}^{r} \| \boldsymbol{Y}\vec{q}_j - \boldsymbol{X}\vec{p}_j \|^2 = \min_{\boldsymbol{P},\boldsymbol{Q}} \sum_{j=1}^{r}\sum_{i=1}^{s} (v_j^i - u_j^i)^2 \quad (8)$$

$$\text{such that } \sum_{j=1}^{r}\sum_{i=1}^{s} \operatorname{sign}(v_j^i,\, u_j^i) = sr \quad (9)$$

After normalizing the samples $\{\vec{y}^i\}_{i=1:s}$ and $\{\vec{x}^i\}_{i=1:s}$ (removing means and dividing by the standard deviation), the sign-correlation constrained optimization problem can be solved by standard canonical correlation analysis (CCA). The CCA problem for the normalized $\{\vec{y}^i\}_{i=1:s}$ and $\{\vec{x}^i\}_{i=1:s}$ is:

$$\max_{\vec{p}_j, \vec{q}_j} \vec{q}_j^{\mathrm{T}} \operatorname{cov}(\boldsymbol{Y}, \boldsymbol{X})\vec{p}_j \quad (10)$$

such that

$$\vec{q}_j^{\mathrm{T}} \operatorname{var}(\boldsymbol{Y},\, \boldsymbol{Y})\vec{q}_j = 1, \quad \vec{p}_j^{\mathrm{T}} \operatorname{var}(\boldsymbol{X},\, \boldsymbol{X})\vec{p}_j = 1 \quad (11)$$

Following the CCA algorithm, the max sign-correlation pair $\vec{p}_1$ and $\vec{q}_1$ are solved first. Then one seeks $\vec{p}_2$ and $\vec{q}_2$ by maximizing the same correlation subject to the constraint that they are to be uncorrelated with the first pair of canonical variables $\vec{w}_1$, $\vec{z}_1$. This procedure is continued until $\vec{p}_r$ and $\vec{q}_r$ are found.

After all $\vec{p}_j$ and $\vec{q}_j$ have been computed, we only need the projection matrix $\boldsymbol{P}$ in $\Delta\vec{x}$ space. We then project each $\Delta\vec{x}^i$ into the sign-correlation subspace to get the reduced feature $\vec{u}^i = \boldsymbol{P}\Delta\vec{x}^i$. Then we partition the whole sample space into independent descent domains by considering the sign of each dimension of $\vec{u}^i$ and group it into the corresponding hyperoctant. Finally, in order to solve Eq. (4) at each iterative step, we learn a descent mapping for every subset at each iterative step with the ridge regression algorithm. When testing a face image, we also use the projection matrix $\boldsymbol{P}$ to find its corresponding descent domain and predict its shape increment at each iterative step.

Regressor-based methods are sensitive to initialization, and sometimes require multiple initializations to produce a stable result [24]. Generally, the obtained results of the landmark positions are accurate and visually plausible when inspected individually, but they may vary drastically on weakly textured areas when the face initialization changes slightly, since in these methods the temporally and spatially coherent nature of videos is not considered. Since we are reconstructing faces from input videos recorded in an uncontrolled environment, the bounding box generated by the face detector can be unstable. The unstable initialization and the sensitive nature of the landmark detector on missing and blurry frames lead to jittery and unconvincing results.

Nevertheless, the set of unstable landmarks is enough to reconstruct a rough facial geometry and texture model of the target person. As in Ref. [17], we first align a generic 3D face mesh to the 2D landmarks. The corresponding indices of the facial landmarks of the nose, eye boundaries, lips, and eyebrow contours are fixed, whereas the vertex indices of the face contour are recomputed with respect to frame specific poses and expressions. To generate uniformly distributed contour points we selectively project possible contour vertices onto the image and sample its convex hull with uniform 2D spacing.

The facial reconstruction problem can be formulated as an optimization problem in which the *pose*, *expression*, and *identity* of the person are determined in a coordinate descent manner.

3.2 Pose estimation

Following Ref. [49] we use a pinhole camera model with radial distortion. Assuming the pixels are square and that the center of projection is coincident with the image center, the projection operation $\prod$ depends on 10 parameters: the 3D orientation R (3×1 vector), the translation t (3×1 vector), the focal length f (scalar), and the distortion parameter k (3×1 vector). We assume the same distortion and focal length for the entire video, and initialize the focal length to be the pixel width of the video and distortion to zero. First, we apply a direct linear transform [50] to estimate the initial rotation and translation then optimize them via the Levenberg–Marquardt method with a robust loss function [51].

The 3D rotation matrix is constructed from the orientation vector R using:

$$\omega \leftarrow R/\sigma, \ \sigma \leftarrow ||R|| \tag{12}$$

$$\cos(\sigma)\mathbb{I}+(1-\cos(\sigma))\nVdash+\sin(\sigma)\begin{vmatrix} 0 & -R_0 & R_1 \\ R_2 & 0 & -R_0 \\ -R_1 & R_0 & 0 \end{vmatrix} \tag{13}$$

whose derivative is computed via forward accumulation automatic differentiation [52].

3.3 Expression estimation

In the pose estimation stage, we used a generic face model for initialization, but to get more accurate results we need to adjust the model according to the expression and identity. We use the FaceWarehouse dataset [43], which contains the performances of 150 people with 47 different expressions. Since we are only tracking facial expressions, we select only the frontal facial vertices because the nose and head shape are not included in the detected landmarks. We flatten the 3D vertices and arrange them into a 3 mode data tensor. We compress the original tensor representing 30k vertices × 150 identities × 47 expressions into a 4k vertices × 50 identities × 25 expression coefficients core using higher order singular value decomposition [53]. Any facial mesh in the dataset can be approximated by the product of its core $B_{\text{exp}} = C \times U_{\text{id}}$ or $B_{\text{id}} = C \times U_{\text{exp}}$, where U_{id} and U_{exp} are the identity and expression orthonormal matrices respectively; B_{exp} is a person with different facial expressions, B_{id} is the same expression performed by different individuals.

For efficiency we first determine the identity with the compressed core and prevent over-fitting with an early stopping strategy. To generate plausible results we need to solve for the uncompressed expression coefficients with early stopping and box constrain them to lie within a valid range, which in the case of FaceWarehouse is between 0 and 1. We do not optimize identity and camera coefficients for individual frames. They are only optimized jointly after expression coefficients have been estimated.

We group the camera parameters into a vector $\theta = [R, t, f]$. We generate a person specific facial mesh B_{id} with this person's identity coefficient I, which results in the same individual performing the 47 defined expressions. The projection operator is defined as $\prod([x,y,z]^{\text{T}}) = r[x,y,z]^{\text{T}} + t$, where r is the 3×3 rotation matrix constructed from Eq. (13) and the radial distortion function $\mathbb{D}$ is defined as

$$\mathbb{D}(X', k) = f \times X'(1 + k_1 r^2 + k_2 r^4) \tag{14}$$

$$\mathbb{D}(Y', k) = f \times Y'(1 + k_1 r^2 + k_2 r^4) \tag{15}$$

$$r^2 = X'^2 + Y'^2, \quad X' = X/Z, \quad Y' = Y/Z \tag{16}$$

$$[X, Y, Z]^{\text{T}} = \prod([x, y, z]^{\text{T}}) \tag{17}$$

We minimize the squared distance between the 2D landmarks L after applying radial distortion while fixing the identity coefficient and pose parameters $\mathbb{D}$:

$$\min_E \frac{1}{2}|L - \mathbb{D}(\prod(B_{\text{id}} \cdot E, \theta), k)|^2 \tag{18}$$

To solve this problem efficiently, we apply the reverse distortion to L, then rotate and translate the vertices. By denoting the projected coordinates by p, the derivative of E can be expressed efficiently as

$$(L - f \cdot p)\left(f \cdot \frac{B^{(i)}_{\text{id}(0,1)} + B^{(i)}_{\text{id}(2)} \cdot p}{Z}\right) \tag{19}$$

We use the Levenberg–Marquardt method for initialization and perform line search [54] to constrain E to lie within the valid range.

3.4 Identity adaption

Since we cannot apply a generic B_{id} to different individuals with differing facial geometry, we solve for the subject's identity in a similar fashion to the expression coefficient. With the estimated expression coefficients from the last step, we generate facial meshes of different individuals performing the estimated expressions. Unlike expression coefficient estimation, we need to solve identity coefficient jointly across I frames with different poses and expressions. We denote the nth facial mesh by B^n_{exp} and minimize the distance:

$$\min_I \sum_n \frac{1}{2}|L^n - \mathbb{D}(\prod(B^n_{\text{exp}} \cdot I, \theta), k)|^2 \tag{20}$$

while fixing all other parameters. Here it is important to exclude inaccurate single frames from being considered otherwise they lead to erroneous identity.

3.5 Camera estimation

Some videos may be captured with camera distortions. In order to reconstruct the 3D facial geometry as accurately as possible, we undistort the video by estimating its focal length and distortion parameters. All of the following dense tracking is performed in undistorted camera space. To avoid local minima caused by over-fitting the distortion parameters, we solve for focal length analytically using:

$$f = \frac{\sum_n L^n}{\sum_n \mathbb{D}(\prod(B^n_{\text{exp}} \cdot I, \theta), k)} \tag{21}$$

then use nonlinear optimization to solve for radial distortion. We find the camera parameters by jointly minimizing the difference between the selected 2D landmarks L and their corresponding projected vertices:

$$\min_k \sum_n \frac{1}{2}|L^n - \mathbb{D}(\prod(B^n_{\text{exp}} \cdot I, \theta), k)|^2 \tag{22}$$

3.6 Average texture estimation

In order to estimate an average texture, we extract per pixel color information from the video frames. We use the texture coordinates provided in FaceWarehouse to normalize the facial texture onto a flattened 2D map. By performing visibility tests we filter out invisible pixels. Since the eyeball and inside of the mouth are not modeled by facial landmarks or FaceWarehouse, we consider their texture separately. Although varying expressions, pose, and lighting conditions lead to texture variation across different frames, we use their summed average as a low rank approximation. Alternatively, we could use the median pixel values as it leads to sharper texture, but at the coarse reconstruction we choose not to because computing the median requires all the images to be available whereas the average can be computed on-the-fly without additional memory costs. Moreover, while the detected landmarks are not entirely accurate, robustness is more important than accuracy. Instead, we selectively compute the median of high quality frames from dense reconstruction to generate better texture in the next stage.

The idea of tracking the facial landmarks by minimizing the difference between synthesized view and the real image is similar to that used in active appearance models (AAM) [3]. The texture variance can be modeled and approximated by principle component analysis, and expression–pose specific texture can be used for better performance. Experimental results show that high rank approximation leads to unstable results because of the landmark detection in-the-wild issues. Moreover, AAM typically has to be trained on manually labeled images that are very accurate. Although it is able to fit the test image with better texture similarity, it is not suitable for robust automated landmark detection. A comparison of our method with traditional AAM method is shown later and examples of failed detections are shown in Fig. 3.

Up to this point, we have been optimizing the 3D coordinates of the facial mesh and the camera parameters. Due to the limited expressiveness of the facial dataset, which only contains 150 persons, the

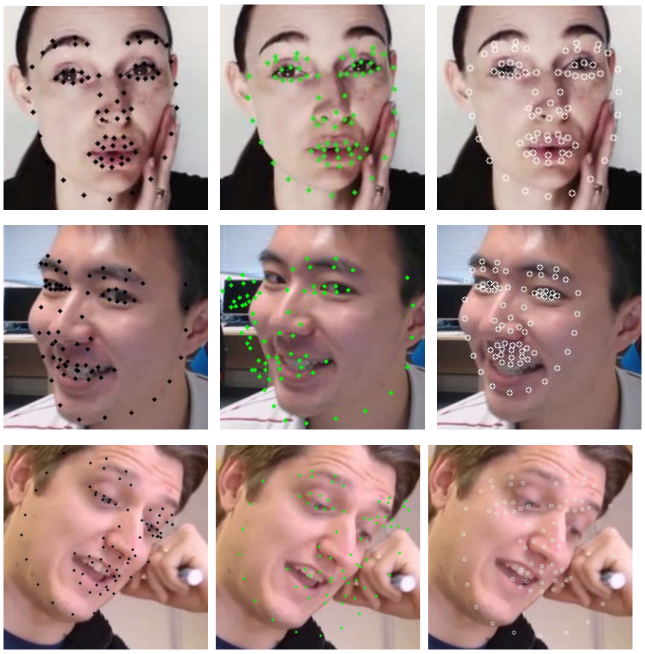


Fig. 3 Landmark tracking comparison. From left to right: ours, in-the-wild, AAM.

fitted facial mesh might not exactly fit the detected landmarks. To increase the expressiveness of the reconstructed model and add more person specific details, we use the method in Ref. [55] to deform the facial mesh reconstructed for each frame. We first assign the depth of the 2D landmarks to that of their corresponding 3D vertices, then unproject them into 3D space. Finally, we use the unprojected 3D coordinates as anchor points to deform the facial mesh of every frame.

Since the deformed facial mesh may not be represented by the original data, we need to add them into the person specific facial meshes B_{exp} and keep the original expression coefficients. Given an expression coefficient E we could reconstruct its corresponding facial mesh $F = B_{\text{exp}}E$. Thus the new deformed mesh base should be computed via $F_{\text{d}} = B_{\text{d}}E_{\text{d}}$. We flatten the deformed and original facial meshes using B_{exp}, then concatenate them together as $B_{\text{c}} = [B; B_{\text{d}}]^{\text{T}}$. We concatenate coefficients of the 47 expressions in FaceWarehouse and the recovered expressions from the video frames as $E_{\text{c}} = [E; E_{\text{d}}]^{\text{T}}$. The new deformed facial mesh base is computed from $B_{\text{d}} = E_{\text{c}}^{-1}B_{\text{c}}$.

We simply compute for each pixel the average color value and run the k-means algorithm [56] on the extracted eyeball and mouth interior textures, saving a few representative k-means centers for fitting different expressions and eye movements. An example of the reconstructed average face texture is shown in Fig. 4(a).

4 Dense reconstruction to refine landmarks

4.1 Face tracking flow

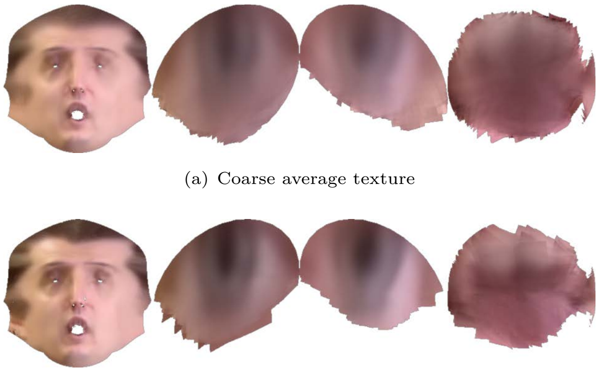
In the previous step we reconstructed an average face model with a set of coarse facial landmarks. To deliver convincing results we need to track and reconstruct all of the vertices even in weakly textured areas. To robustly capture the 3D facial performance in each frame, we formulate the problem in terms of 3D optical flow and solve for dense correspondence between the 3D model and each video frame, optimally deforming the reference mesh to fit the seen image. We use the rendered average shape as initialization and treat it as the previous frame; we use the real image as the current frame to densely compute the displacement of all vertices. Assuming the pixel intensity does not change by the displacement, we may write:

$$I(x, y) = C(x + u, y + v) \tag{23}$$

where I denotes the intensity value of the rendered image, C the real image, and x and y denote pixel coordinates. In addition, the gradient value of each pixel should also not change due to displacement because not only the pixel intensity but also the texture stay the same, which can be expressed as

$$\nabla I(x, y) = \nabla C(x + u, y + v) \tag{24}$$

Finally, the smoothness constraint dictates that pixels should stay in the same spatial arrangement



(a) Coarse average texture

(b) Dense average texture

Fig. 4 Refined texture after robust dense tracking.

to their original neighbors to avoid the aperture problem, especially since many facial areas are weakly textured, i.e., have no strong gradient. We search for $f = (u, v)^{\mathrm{T}}$ that satisfies the pixel intensity, gradient, and smoothness constraints.

By denoting each projected vertex of the face mesh by $p = \mathbb{D}(\prod(B^n_{\mathrm{id}} \cdot E, \theta), k)$, we formulate the energy as

$$E_{\text{flow}}(f) = \sum_v |I(p+f) - C(p)|^2 + \alpha(|\nabla f|^2) + \beta(|\partial f|^2) \tag{25}$$

Here $|\nabla f|^2$ is a smoothness term and $\beta(|\partial f|^2)$ is a piecewise smooth term. As this is a highly nonlinear problem we adopt the numerical approximation in Ref. [57] and take a multi-scale approach to achieve robustness. We do not use the additional match term Eq. (26) in Ref. [58], where $\gamma(p)$ is the match weight: although we have the match from the landmarks to the vertices, we cannot measure the quality of the landmarks, as well as the matches, so:

$$E_{\text{match}}(f) = \sum_p \mu(p) |p_I + f - p_C|^2 \mathrm{d}p \tag{26}$$

4.2 Robust tracking

Standard optical flow suffers from drift, occlusion, and varying visibility because of lack of explicit modeling. Since we already have a rough prior of the face from the coarse reconstruction step, we use it to correct and regularize the estimated optical flow.

We test the visibility of each vertex by comparing its transformed value to its rendered depth value. If it is larger than a threshold then it is considered to be invisible and not used to solve for pose and expression coefficient. To detect partially occluded areas we compute both the forward flow (rendered to real image f_{f}) and backward flow (real image to rendered f_{b}), and compute the difference for each of the vertices' projections:

$$\sum_p |f_{\mathrm{f}}(p) + f_{\mathrm{b}}(p + f_{\mathrm{f}}(p))|^2 \tag{27}$$

We use the GPU to compute the flow field whereas the expression coefficient and pose are computed on the CPU. Solving them for all vertices can be expensive when there is expression and pose variation, so to reduce the computational cost, we also check the norm of $f_{\mathrm{f}}(p)$ to filter out pixels with negligible displacement.

Because of the piecewise smoothness constraint, we consider vertices with large forward and backward flow differences to be occluded and exclude them from the solution process. We first find the rotation and translation, then the expression coefficients after putative flow fields have been identified. The solution process is similar to that used in the previous section with the exception that we update each individual vertex at the end of the iterations to fit the real image as closely as possible. To exploit temporal and spatial coherence, we use the average of a frame's neighboring frames to initialize its pose and expression, then update them using coordinate descent. If desired, we reconstruct the average face model and texture from the densely tracked results and use the new model and texture to perform robust tracking again. An example of updated reconstructed average texture is shown in Fig. 4, which is *sharper* and *more accurate* than the coarsely reconstructed texture. Filtered vertices and the tracked mesh are shown in Fig. 5, where putative vertices are color coded and filtered out vertices are hidden. Note that the color of the actress' hand is very close to that of her face, so it is hard to mask out by color difference thresholding without piecewise smoothness regularization.

4.3 Texture update

Finally, after robust dense tracking results and the validity of each vertex have been determined, each valid vertex can be optionally optimized individually to recover further details. This is done in a coordinate descent manner with respect to the pose parameters. Updating all vertices with

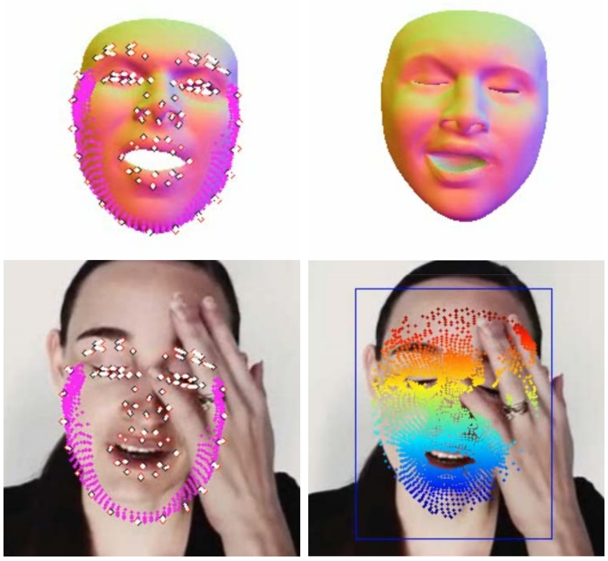


Fig. 5 Example of reconstruction with occlusion.

a standard nonlinear optimization routine might be inefficient because of the computational cost of inverting or approximating a large second order Hessian matrix, which is sparse in this case because the points do not have influence on each other. Thus, instead, we use the Schur complement trick [59] to reduce the computational cost. The whole pipeline of our method is summarized in Algorithm 1. Convergence is determined by the norm of the optical flow displacement. This criterion indicates whether further vertex adjustment is possible or necessary to minimize the difference between the observed image and synthesized result.

Compared to the method in Ref. [19], which also formulates the face tracking problem in an optical flow context, our method is more robust. In videos with large pose and expression variation, inaccurate coarse facial landmark initialization and partial occlusion caused by texturally similar objects, our method is more accurate and expressive and generates smoother results than the coarse reconstruction computed with landmarks from in-the-wild methods in Ref. [30].

Algorithm 1: Automatic dense facial capture

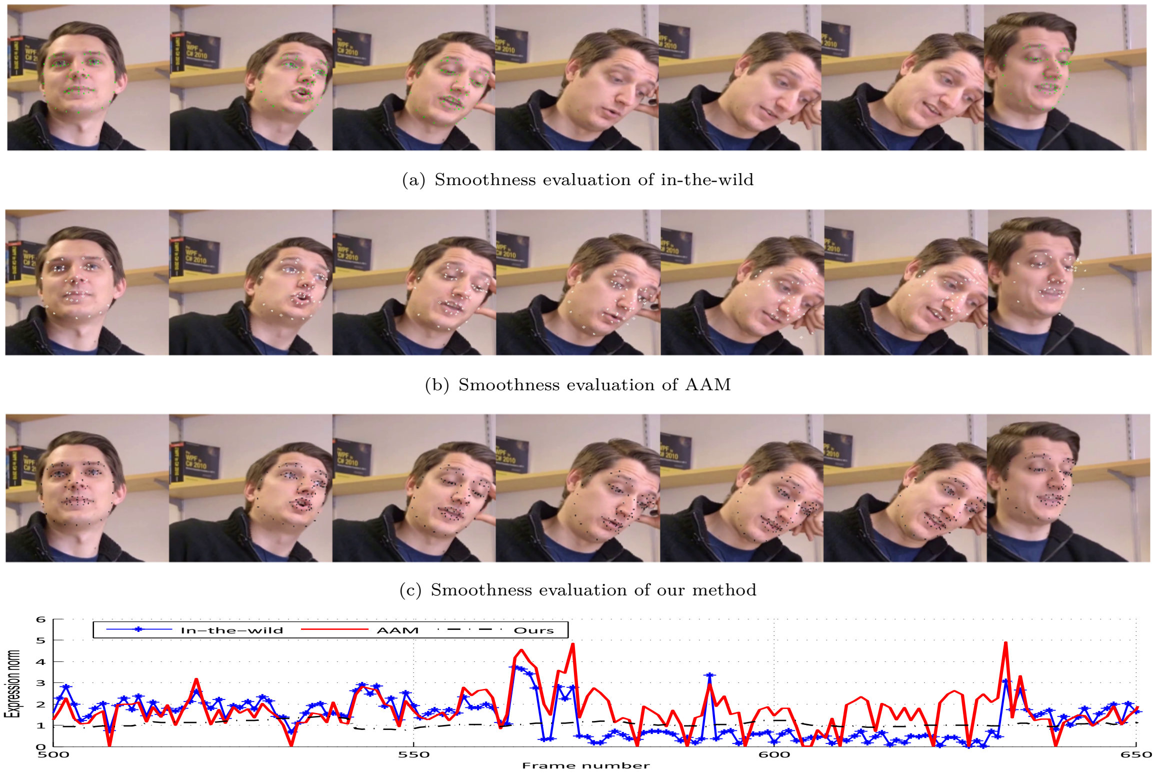
```
Input: Video
CCA-GSDM landmark detection
Solve Pose on landmarks
Solve Expression using Eq. (18) on landmarks
Solve Identity using Eq. (20) on landmarks
Solve Focal using Eq. (21) on landmarks
Solve Distortion using Eq. (22) on landmarks
while not converged do
  while norm(flow) > threshold do
    Determine vertex validity using depth check
    Determine vertex validity using Eq. (27)
    Determine vertex validity using norm of flow displacement
    Solve Pose on optical flow
    Solve Expression using Eq. (18) on optical flow
    if Inner max iteration reached then
      break
    end if
  end while
  Update camera
  Update vertex
  Update texture
  if Outer max iteration reached then
    break
  end if
end while
Output: Facial meshes, poses, expressions
```

5 Experiments

Our proposed method aims to deliver smooth facial performances and landmark tracking in uncontrolled in-the-wild videos. Although recently a new dataset has been introduced designed for facial landmark tracking in the wild [34], it is not adequate for this work since we aim to deliver *smooth* tracking results rather than just locating landmark positions. In addition, we also concentrate on capturing detail to reconstruct realistic expressions. Comparison of the expression norm between the coarse landmarks and dense tracking is shown in Fig. 6.

In order to evaluate the performance of our robust method, AAM [3, 22], and an in-the-wild regressor-based method [28, 30] working as *fully automated* methods, we collected 50 online videos with frame counts ranging from 150 to 897 and manually labeled them. Their resolution is 640 × 360. There are a wide range of different poses and expressions in these videos, and heavy partial occlusion as well. Being *fully automated* means that given any in-the-wild video no more additional effort is required to tune the model. We manually label landmarks for a quarter of the frames sampled uniformly throughout the entire video to train a person specific AAM model then use the trained model to track the landmarks. Note that doing so *disqualifies* the AAM approach as a *fully automated* method. Next we manually correct the tracked result to generate a smooth and visually plausible landmark sequence. We treat such sequences as ground truth and test each method's accuracy against it. We also use these manually labeled landmarks to build corresponding coarse facial models and texture in a similar way to the approach used in Section 3. The result is shown in Table 1. Each numeric column represents the error between the ground truth and the method's output. Following standard practice [24, 28, 60], we use the inter-pupillary distance normalized landmark error. Mesh reconstruction error is measured by the average L_2 distance between the reconstructed meshes. Texture error is measured by the average of per-pixel color difference between the reconstructed textures.

We mainly compare our method to appearance-based methods [3, 22] and in-the-wild methods [28, 30] because they are appropriate for in-the-wild video and have similar aims to minimize texture


(a) Smoothness evaluation of in-the-wild

(b) Smoothness evaluation of AAM

(c) Smoothness evaluation of our method

(d) Overall smoothness evaluation

Fig. 6 Example tracking results.

Table 1 Whole set error comparison

Method	Mesh	Texture	Landmark
Ours 1 iteration	**13.3**	**29.2**	**4.4**
Ours 2 iteration	**10.3**	**25.4**	**3.2**
Kazemi and Sullivan [30]	33.2	95.4	9.7
Ren et al. [28]	37.8	114.3	7.4
Donner et al. [22]	23.3	67.7	15.2
Cootes et al. [3]	24.3	86.5	24.4
Low	41.3	136.8	35.4
High	54.3	186.5	32.2

discrepancy between synthetic views and real images. We have also built a CUDA-based face tracking application using our method; it can achieve real-time tracking. The tested video resolution is 640 × 360, for which ir achieves more than 30 fps, benefiting from CUDA speed up. The dense points (there are 5760 of them) are from the frontal face of a standard blendshape mesh.

For completeness we also used the detected landmarks obtained from in-the-wild methods to train the AAM models, then used these to detect landmarks in videos. Doing so *qualifies* them as *fully automated* methods again. Due to the somewhat inconsistent results produced by in-the-wild landmark detectors, we use both high and low rank texture approximation thresholds when training the AAM. Note that although Donner et al. [22] propose use of regression relevant information which may be discarded by purely generative PCA-based models, they also use an approximate texture variance model. Models trained with low rank variance are essentially the same as our approach of just taking the average of all images. While low rank AAM can accurately track the pose of the face most of the time when there is no large rotation, it fails to track facial point movements such as closing and opening of eyes, and talking, because the low rank model limits its expressiveness. High rank AAM, on the other hand, can track facial point movements but produces unstable results due to the instability of the training data provided by the in-the-wild method. Experimental results of training AAM with landmarks detected by the method in Ref. [30] are shown in the *Low* and *High* columns of Table 1

We also considered spearately a challenging subset of the videos, in which there is more partial occlusion, large head rotation or exaggerated facial expression. The performance of each method is given in Table 2. A comparison of our method to AAM and the in-the-wild method is shown in Fig. 6, where

Table 2 Challenging subset error comparison

Method	Mesh	Texture	Landmark
Ours 1 iteration	**41.7**	**59.1**	**7.2**
Ours 2 iteration	**15.1**	**35.2**	**4.1**
Kazemi and Sullivan [30]	92.3	95.4	19.2
Ren et al. [28]	88.1	114.3	11.4
Donner et al. [22]	97.3	142.7	21.9
Cootes et al. [3]	87.9	136.2	21.3
Low	114.3	146.5	25.3
High	134.7	186.2	33.4

the x axis is the frame count and the y axis is the norm of the expression coefficient. Compared to facial performance tracking with only coarse and inaccurate landmarks, our method is very stable and has a lower error rate than the other two methods. Further landmark tracking results are shown in Fig. 7. Additional results and potential applications are shown in the Electronic Supplementary Material.

6 Conclusions

We have proposed a novel fully automated method for robust facial landmark detection and tracking across poses and expressions for in-the-wild monocular videos. In our work, shape-face canonical correlation analysis is combined with a global supervised descent method to achieve robust coarse 2D facial landmark detection across poses. We perform coarse-to-dense 3D facial expression reconstruction with a 3D facial prior to boost tracked landmarks. We have evaluated its performance with respect to state-of-the-art landmark detection methods and empirically compared the tracked results to those of conventional approaches. Compared to conventional tracking methods that are able to capture subtle facial movement details, our method is fully automated, just as expressive and robust in noisy situations. Compared to other robust in-the-wild methods, our method delivers smooth tracking results and is able to capture small facial movements even for weakly textured areas. Moreover, we can accurately compute the possibility of a facial area being occluded in a particular frame, allowing us to avoid erroneous results. The 3D facial geometry and performance reconstructed and captured by our method are not only accurate and visually convincing, but we can also extract 2D landmarks from the mesh and use them in other methods that depend on 2D facial landmarks, such as facial editing, registration, and recognition.

Currently we are only using the average texture model for all poses and expressions. To further improve the expressiveness, we could adopt a similar approach to that taken for active appearance models, where after we have robustly built an average face model, texture variance caused by different lighting conditions, pose and expression variation could also be modeled to improve the expressiveness and accuracy of the tracking results.

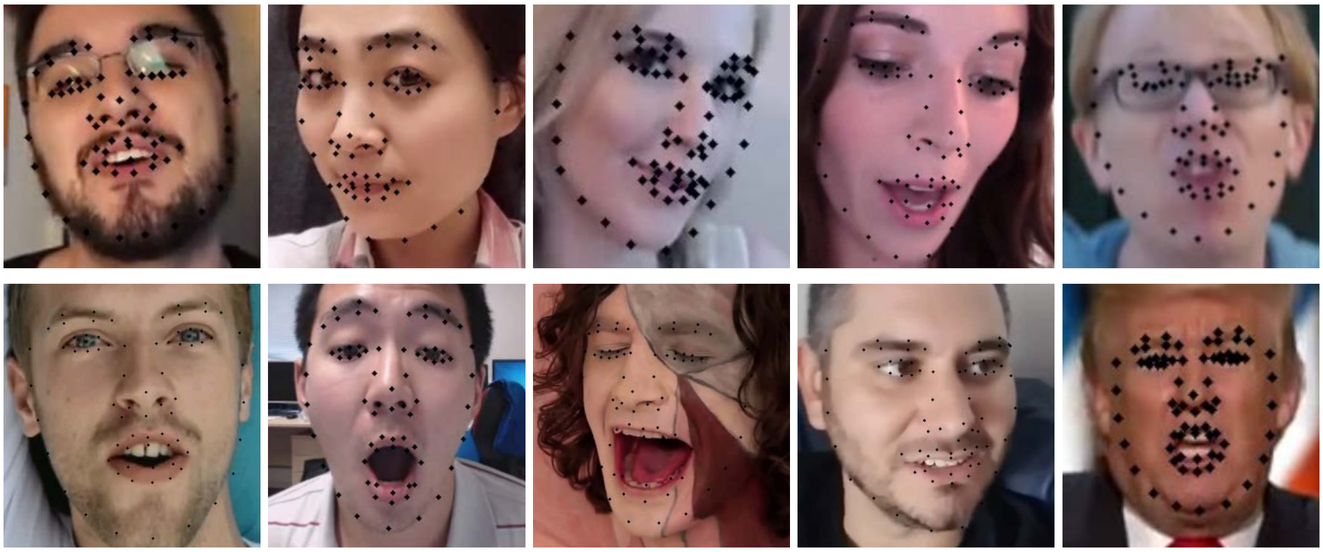


Fig. 7 Landmark tracking results.

Acknowledgements

This work was supported by the Harbin Institute of Technology Scholarship Fund 2016 and the National Centre for Computer Animation, Bournemouth University.

Electronic Supplementary Material Supplementary material is available in the online version of this article at http://dx.doi.org/10.1007/s41095-016-0068-y.

References

[1] Mori, M.; MacDorman, K. F.; Kageki, N. The uncanny valley [from the field]. *IEEE Robotics & Automation Magazine* Vol. 19, No. 2, 98–100, 2012.

[2] Cootes, T. F.; Taylor, C. J.; Cooper, D. H.; Graham, J. Active shape models—Their training and application. *Computer Vision and Image Understanding* Vol. 61, No. 1, 38–59, 1995.

[3] Cootes, T. F.; Edwards, G. J.; Taylor, C. J. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 23, No. 6, 681–685, 2001.

[4] Cristinacce, D.; Cootes, T. F. Feature detection and tracking with constrained local models. In: Proceedings of the British Machine Conference, 95.1–95.10, 2006.

[5] Gonzalez-Mora, J.; De la Torre, F.; Murthi, R.; Guil, N.; Zapata, E. L. Bilinear active appearance models. In: Proceedings of IEEE 11th International Conference on Computer Vision, 1–8, 2007.

[6] Lee, H.-S.; Kim, D. Tensor-based AAM with continuous variation estimation: Application to variation-robust face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 31, No. 6, 1102–1116, 2009.

[7] Cao, X.; Wei, Y.; Wen, F.; Sun, J. Face alignment by explicit shape regression. U.S. Patent Application 13/728,584. 2012-12-27.

[8] Xiong, X.; De la Torre, F. Supervised descent method and its applications to face alignment. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 532–539, 2013.

[9] Xing, J.; Niu, Z.; Huang, J.; Hu, W.; Yan, S. Towards multi-view and partially-occluded face alignment. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1829–1836, 2014.

[10] Yan, J.; Lei, Z.; Yi, D.; Li, S. Z. Learn to combine multiple hypotheses for accurate face alignment. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, 392–396, 2013.

[11] Burgos-Artizzu, X. P.; Perona, P.; Dollár, P. Robust face landmark estimation under occlusion. In: Proceedings of the IEEE International Conference on Computer Vision, 1513–1520, 2013.

[12] Yang, H.; He, X.; Jia, X.; Patras, I. Robust face alignment under occlusion via regional predictive power estimation. *IEEE Transactions on Image Processing* Vol. 24, No. 8, 2393–2403, 2015.

[13] Feng, Z.-H.; Huber, P.; Kittler, J.; Christmas, W.; Wu, X.-J. Random cascaded-regression copse for robust facial landmark detection. *IEEE Signal Processing Letters* Vol. 22, No. 1, 76–80, 2015.

[14] Yang, H.; Jia, X.; Patras, I.; Chan, K.-P. Random subspace supervised descent method for regression problems in computer vision. *IEEE Signal Processing Letters* Vol. 22, No. 10, 1816–1820, 2015.

[15] Zhu, S.; Li, C.; Loy, C. C.; Tang, X. Face alignment by coarse-to-fine shape searching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4998–5006, 2015.

[16] Cao, C.; Hou, Q.; Zhou, K. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Transactions on Graphics* Vol. 33, No. 4, Article No. 43, 2014.

[17] Liu, S.; Yang, X.; Wang, Z.; Xiao, Z.; Zhang, J. Real-time facial expression transfer with single video camera. *Computer Animation and Virtual Worlds* Vol. 27, Nos. 3–4, 301–310, 2016.

[18] Tzimiropoulos, G.; Pantic, M. Optimization problems for fast AAM fitting in-the-wild. In: Proceedings of the IEEE International Conference on Computer Vision, 593–600, 2013.

[19] Suwajanakorn, S.; Kemelmacher-Shlizerman, I.; Seitz, S. M. Total moving face reconstruction. In: *Computer Vision–ECCV 2014*. Fleet, D.; Pajdla, T.; Schiele, B.; Tuytelaars, T. Eds. Springer International Publishing, 796–812, 2014.

[20] Cootes, T. F.; Taylor, C. J. Statistical models of appearance for computer vision. 2004. Available at http://personalpages.manchester.ac.uk/staff/timothy.f.cootes/Models/app_models.pdf.

[21] Yan, S.; Liu, C.; Li, S. Z.; Zhang, H.; Shum, H.-Y.; Cheng, Q. Face alignment using texture-constrained active shape models. *Image and Vision Computing* Vol. 21, No. 1, 69–75, 2003.

[22] Donner, R.; Reiter, M.; Langs, G.; Peloschek, P.; Bischof, H. Fast active appearance model search using canonical correlation analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 28, No. 10, 1690–1694, 2006.

[23] Matthews, I.; Baker, S. Active appearance models revisited. *International Journal of Computer Vision* Vol. 60, No. 2, 135–164, 2004.

[24] Cao, X.; Wei, Y.; Wen, F.; Sun, J. Face alignment by explicit shape regression. *International Journal of Computer Vision* Vol. 107, No. 2, 177–190, 2014.

[25] Dollár, P.; Welinder, P.; Perona, P. Cascaded pose regression. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1078–1085, 2010.

[26] Zhou, S. K.; Comaniciu, D. Shape regression machine. In: *Information Processing in Medical Imaging.* Karssemeijer, N.; Lelieveldt, B. Eds. Springer Berlin Heidelberg, 13–25, 2007.

[27] Burgos-Artizzu, X. P.; Perona, P.; Dollár, P.

Robust face landmark estimation under occlusion. In: Proceedings of the IEEE International Conference on Computer Vision, 1513–1520, 2013.

[28] Ren, S.; Cao, X.; Wei, Y.; Sun, J. Face alignment at 3000 fps via regressing local binary features. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1685–1692, 2014.

[29] Cootes, T. F.; Ionita, M. C.; Lindner, C.; Sauer, P. Robust and accurate shape model fitting using random forest regression voting. In: *Computer Vision–ECCV 2012*. Fitzgibbon, A.; Lazebnik, S.; Perona, P.; Sato, Y.; Schmid, C. Eds. Springer Berlin Heidelberg, 278–291, 2012.

[30] Kazemi, V.; Sullivan, J. One millisecond face alignment with an ensemble of regression trees. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1867–1874, 2014.

[31] Sagonas, C.; Tzimiropoulos, G.; Zafeiriou, S.; Pantic, M. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, 397–403, 2013.

[32] Zhou, F.; Brandt, J.; Lin, Z. Exemplar-based graph matching for robust facial landmark localization. In: Proceedings of the IEEE International Conference on Computer Vision, 1025–1032, 2013.

[33] Huang, G. B.; Ramesh, M.; Berg, T.; Learned-Miller, E. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[34] Shen, J.; Zafeiriou, S.; Chrysos, G. G.; Kossaifi, J.; Tzimiropoulos, G.; Pantic, M. The first facial landmark tracking in-the-wild challenge: Benchmark and results. In: Proceedings of the IEEE International Conference on Computer Vision Workshop, 1003–1011, 2015.

[35] Cao, C.; Bradley, D.; Zhou, K.; Beeler, T. Realtime high-fidelity facial performance capture. *ACM Transactions on Graphics* Vol. 34, No. 4, Article No. 46, 2015.

[36] Cao, C.; Wu, H.; Weng, Y.; Shao, T.; Zhou, K. Real-time facial animation with image-based dynamic avatars. *ACM Transactions on Graphics* Vol. 35, No. 4, Article No. 126, 2016.

[37] Garrido, P.; Valgaerts, L.; Wu, C.; Theobalt, C. Reconstructing detailed dynamic face geometry from monocular video. *ACM Transactions on Graphics* Vol. 32, No. 6, Article No. 158, 2013.

[38] Ichim, A. E.; Bouaziz, S.; Pauly, M. Dynamic 3D avatar creation from hand-held video input. *ACM Transactions on Graphics* Vol. 34, No. 4, Article No. 45, 2015.

[39] Saito, S.; Li, T.; Li, H. Real-time facial segmentation and performance capture from RGB input. *arXiv preprint* arXiv:1604.02647, 2016.

[40] Shi, F.; Wu, H.-T.; Tong, X.; Chai, J. Automatic acquisition of high-fidelity facial performances using monocular videos. *ACM Transactions on Graphics* Vol. 33, No. 6, Article No. 222, 2014.

[41] Thies, J.; Zollhöfer, M.; Stamminger, M.; Theobalt, C.; Nießner, M. Face2face: Real-time face capture and reenactment of RGB videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1, 2016.

[42] Furukawa, Y.; Ponce, J. Accurate camera calibration from multi-view stereo and bundle adjustment. *International Journal of Computer Vision* Vol. 84, No. 3, 257–268, 2009.

[43] Cao, C.; Weng, Y.; Zhou, S.; Tong, Y.; Zhou, K. FaceWarehouse: A 3D facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics* Vol. 20, No. 3, 413–425, 2014.

[44] Newcombe, R. A.; Izadi, S.; Hilliges, O.; Molyneaux, D.; Kim, D.; Davison, A. J.; Kohi, P.; Shotton, J.; Hodges, S.; Fitzgibbon, A. KinectFusion: Realtime dense surface mapping and tracking. In: Proceedings of the 10th IEEE International Symposium on Mixed and Augmented Reality, 127–136, 2011.

[45] Weise, T.; Bouaziz, S.; Li, H.; Pauly, M. Realtime performance-based facial animation. *ACM Transactions on Graphics* Vol. 30, No. 4, Article No. 77, 2011.

[46] Blanz, V.; Vetter, T. A morphable model for the synthesis of 3D faces. In: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, 187–194, 1999.

[47] Yan, J.; Zhang, X.; Lei, Z.; Yi, D.; Li, S. Z. Structural models for face detection. In: Proceedings of the 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition, 1–6, 2013.

[48] Xiong, X.; De la Torre, F. Global supervised descent method. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2664–2673, 2015.

[49] Snavely, N. Bundler: Structure from motion (SFM) for unordered image collections. 2010. Available at http://www.cs.cornell.edu/∼snavely/bundler/.

[50] Chen, L.; Armstrong, C. W.; Raftopoulos, D. D. An investigation on the accuracy of three-dimensional space reconstruction using the direct linear transformation technique. *Journal of Biomechanics* Vol. 27, No. 4, 493–500, 1994.

[51] Moré, J. J. The Levenberg–Marquardt algorithm: Implementation and theory. In: *Numerical Analysis*. Watson, G. A. Ed. Springer Berlin Heidelberg, 105–116, 1978.

[52] Rall, L. B. *Automatic Differentiation: Techniques and Applications*. Springer Berlin Heidelberg, 1981.

[53] Kolda, T. G.; Sun, J. Scalable tensor decompositions for multi-aspect data mining. In: Proceedings of the 8th IEEE International Conference on Data Mining, 363–372, 2008.

[54] Li, D.-H.; Fukushima, M. A modified BFGS method and its global convergence in nonconvex minimization. *Journal of Computational and Applied Mathematics* Vol. 129, Nos. 1–2, 15–35, 2001.

[55] Igarashi, T.; Moscovich, T.; Hughes, J. F. As-rigid-as-possible shape manipulation. *ACM Transactions on Graphics* Vol. 24, No. 3, 1134–1141, 2005.

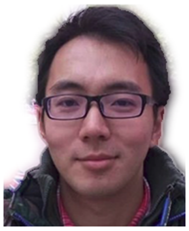
[56] Hartigan, J. A.; Wong, M. A. Algorithm AS 136: A *K*-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* Vol. 28, No. 1, 100–108, 1979.

[57] Brox, T.; Bruhn, A.; Papenberg, N.; Weickert, J. High accuracy optical flow estimation based on a theory for warping. In: *Computer Vision–ECCV 2004*. Pajdla, T.; Matas, J. Eds. Springer Berlin Heidelberg, 25–36, 2004.

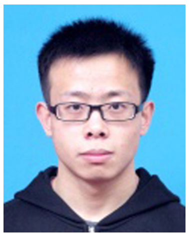
[58] Brox, T.; Malik, J. Large displacement optical flow: Descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 33, No. 3, 500–513, 2011.

[59] Agarwal, S.; Snavely, N.; Seitz, S. M.; Szeliski, R. Bundle adjustment in the large. In: *Computer Vision–ECCV 2010*. Daniilidis, K.; Maragos, P.; Paragios, N. Eds. Springer Berlin Heidelberg, 29–42, 2010.

[60] Belhumeur, P. N.; Jacobs, D. W.; Kriegman, D. J.; Kumar, N. Localizing parts of faces using a consensus of exemplars. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 35, No. 12, 2930–2940, 2013.



Shuang Liu received his B.S. degree in computer science from the Hebei University of Technology, China, in 2014. He is currently a Ph.D. student in the National Centre for Computer Animation, Bournemouth University, UK. His research interests include computer vision and computer animation.



Yongqiang Zhang received his B.S. and M.S. degrees from Harbin Institute of Technology, China, in 2012 and 2014, respectively. He is currently a Ph.D. student in the School of Computer Science and Technology, Harbin Institute of Technology, China. His research interests include machine learning, computer vision, object tracking, and facial animation.

Xiaosong Yang is currently a senior lecturer in the National Centre for Computer Animation (NCCA), Bournemouth University, UK. His research interests include interactive graphics and animation, rendering and modeling, virtual reality, virtual surgery simulation, and CAD. He received his bachelor (1993) and master (1996) degrees in computer science from Zhejiang University, China, and his Ph.D. degree (2000) in computing mechanics from Dalian University of Technology, China. He spent two years as a postdoc (2000–2002) in Tsinghua University working on scientific visualization, and one year (2001–2002) as a research assistant in the Virtual Reality, Visualization and Imaging Research Centre of the Chinese University of Hong Kong. In 2003, he came to NCCA to continue his work on computer animation.

Daming Shi received his Ph.D. degree in mechanical control from Harbin Institute of Technology, China, and Ph.D. degree in computer science from the University of Southampton, UK. He had served as an assistant professor in Nanyang Technological University, Singapore, from 2002. Dr. Shi is currently a chair professor in Harbin Institute of Technology, China. His current research interests include machine learning, medical image processing, pattern recognition, and neural networks.

Jian J. Zhang is a professor of computer graphics in the National Centre for Computer Animation, Bournemouth University, UK, and leads the Computer Animation Research Centre. His research focuses on a number of topics relating to 3D computer animation, including virtual human modelling and simulation, geometric modelling, motion synthesis, deformation, and physics-based animation. He is also interested in virtual reality and medical visualisation and simulation. Prof. Zhang has published over 200 peer reviewed journal and conference publications. He has chaired over 30 international conferences and symposia, and served on a number of editorial boards. Prof. Zhang is also one of the two co-founders of the EPSRC-funded multi-million pound Centre for Digital Entertainment (CDE) with Prof. Phil Willis in the University of Bath.